

# La pertinence des modeles et des formalismes de description des interfaces pour l'évaluation ergonomique

Frédéric Aschehoug

## ► To cite this version:

Frédéric Aschehoug. La pertinence des modeles et des formalismes de description des interfaces pour l'évaluation ergonomique. [Rapport de recherche] RR-1416, INRIA. 1991. inria-00075144

**HAL Id: inria-00075144**

**<https://hal.inria.fr/inria-00075144>**

Submitted on 24 May 2006

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNITÉ DE RECHERCHE  
IRIA-ROCQUENCOURT

Institut National  
de Recherche  
en Informatique  
et en Automatique

Domaine de Voluceau  
Rocquencourt  
BP 105  
78153 Le Chesnay Cedex  
France  
Tél. (1) 39.63 55 11

# Rapports de Recherche

N° 1416

## *Programme 3*

*Intelligence artificielle, Systèmes cognitifs et  
Interaction homme-machine*

## **LA PERTINENCE DES MODELES ET DES FORMALISMES DE DESCRIPTION DES INTERFACES POUR L'EVALUATION ERGONOMIQUE**

**Frédéric ASCHEHOUG**

**Avril 1991**



\* R R - 1 4 1 6 \*



Programme 3

Projet de Psychologie Ergonomique pour l'Informatique

**LA PERTINENCE DES MODELES ET DES FORMALISMES DE  
DESCRIPTION DES INTERFACES POUR L'EVALUATION  
ERGONOMIQUE**

Frédéric Aschehoug

Février 1991

## **La pertinence des modèles et des formalismes de description des interfaces pour l'évaluation ergonomique**

### **Résumé**

Ce rapport porte sur les modèles et les formalismes de description des interfaces homme-ordinateur. On veut construire un modèle de description de l'interface adapté à l'évaluation ergonomique. On commence par introduire des éléments généraux sur les points de vue descriptifs sur les systèmes. Puis on distingue des modèles classificatoires et des formalismes, tous deux spécifiques à la description des interfaces. Les premiers sont considérés comme des outils macroscopiques de description. Les seconds portent plus sur le détail des composants des premiers. Au fur et à mesure de la présentation des travaux, nous sélectionnons les caractéristiques utiles pour tendre vers la construction du modèle visé. On en arrive à la conclusion que le formalisme du graphe de transition appliqué au comportement des objets de l'interface enrichi d'une grammaire sémantique pour décrire les transformations sur ces objets pourrait être un cadre de formalisation à l'intérieur du modèle.

**Mots-clés :** Interface ergonomique - Modèles et formalismes - Evaluation.

### **Usefulness of models and interface description formalisms for ergonomic evaluation**

#### **Abstract**

This report concerns the models and description formalisms for user interfaces. The goal is to design an interface description model adapted to the ergonomic evaluation. First, a number of general considerations are presented about the description of man-machine systems. Then, for the description of interfaces, a distinction is introduced between classification models and formal description models. The first type of models can be considered as "general" description tools, while the second type concern more detailed aspects. Along the description of the various models, a set of usable characteristics of the models are selected with the goal of building a desirable ergonomic model. The major features of the candidate model include the application to the interface objects of a general transition network and added semantic features under a semantic grammar used to describe the various objects transformations and transitions.

**Keywords :** Ergonomic interface, Models and formalisms, Evaluation.

## SOMMAIRE

1	Introduction .....	1
1	Les variétés de points de vue descriptifs sur les systèmes.....	2
1.1	Approche causale et approche fonctionnelle .....	2
1.2	La description du système en termes de modèles mentaux.....	5
2	Les descriptions qui ont une fonction classificatoire sur les éléments de l'interface.....	7
2.1	Les descriptions fondées sur les niveaux d'interface .....	7
2.2	Les descriptions des connaissances requises pour utiliser l'interface .....	11
	a) La description des connaissances sémantiques .....	11
	b) La description des connaissances procédurales requises pour atteindre un but .....	14
	c) La description du modèle conceptuel de l'interface.....	15
3	Les descriptions centrées sur le dialogue et fondées sur une décomposition fonctionnelle de l'interface .....	17
3.1	La décomposition de l'interface en composants fonctionnels.....	17
3.2	La description du composant contrôle du dialogue.....	18
	a) l'utilisation des grammaires B.N.F .....	18
	b) Les graphes de transition.....	20
4	Les descriptions composites non formelles .....	21
5	Conclusions et prolongements .....	24
	Références bibliographiques .....	26
	Index des auteurs .....	33

## **Introduction**

### **Position du problème**

Ce travail porte sur les méthodes d'évaluation et de conception des interfaces homme/ordinateur. On cherche à construire un modèle de l'interface propre à l'activité d'évaluation. Pour répondre à cet objectif, nous nous sommes livrés à un examen bibliographique afin de faire le point sur les approches actuelles de description d'interfaces. Il convenait dans cette démarche d'examiner les techniques de représentation de l'interface afin d'évaluer leur pertinence pour l'ergonomie. Ce travail est un aspect d'une question plus fondamentale dont elle est en partie dépendante, celle de l'étude de l'activité<sup>1</sup> mentale d'évaluation d'interface. Il n'est pas question, ici, de l'activité mentale d'évaluation. On se bornera à faire un examen bibliographique en tâchant au fur et à mesure de la présentation d'apporter un jugement sur l'intérêt des différentes méthodes de description de l'interface pour l'évaluation ergonomique.

La question de la description de l'interface est ancienne. Depuis longtemps, des méthodes de spécification d'interface sont disponibles en informatique. Toutefois ces méthodes n'ont pas été spécifiquement conçues afin d'intégrer les contraintes ergonomiques. La spécificité de notre travail est de viser à construire un modèle de l'interface qui permette conjointement d'intégrer les règles ergonomiques de conception et de répondre aux exigences de l'activité d'évaluation de l'interface.

L'idée directrice du travail peut être formulée en ces termes. Tout au long du processus de conception une interface est décrite sous différents points de vue. L'application de ces points de vue aboutit à une suite de descriptions relatives au même objet. Or l'évaluation de cet objet repose en partie sur des informations dispersées dans les différentes représentations construites. Nous cherchons à substituer à cet éparpillement une représentation unifiée qui intègre les informations pertinentes pour l'évaluation ergonomique.

### **Contexte de l'étude**

Ce travail s'intègre dans un programme de recherche qui vise à contribuer aux spécifications d'un Système de Gestion des Interfaces Utilisateur<sup>2</sup>. Ce type de système se développe de manière intensive depuis peu de temps. Une propriété

---

<sup>1</sup> Sur cette question le lecteur pourra consulter Pollier (1990).

<sup>2</sup> Pour avoir un panorama exhaustif de l'état de l'art sur ces systèmes, le lecteur consultera le texte de El Mrabet (1991). Signalons également l'état de l'art de Löwgren (1988).

remarquable de ces systèmes est d'offrir la possibilité de concevoir l'interface séparément de l'application. Cette possibilité technique d'indépendance de l'interface a actualisé la question de l'introduction d'un composant d'aide au diagnostic ergonomique au sein des U.I.M.S. Hartson et Hix (1989a). C'est ainsi qu'une des priorités fonctionnelles de ce composant est, à moyen terme, la production d'une description ergonomique de l'interface.

## 1 Les variétés de points de vue descriptifs sur les systèmes

Plusieurs travaux ont fait des propositions en vue de définir les différents points de vue par lesquels on peut aborder la description d'un système. Ces propositions ont l'intérêt pour nous de constituer un cadre d'analyse pour la description des interfaces. Qu'en est-il de ces propositions ?

### 1.1 Approche causale et approche fonctionnelle

Rasmussen (1985) distingue deux manières de décrire un système. La première consiste à décrire le système à partir de l'énumération des fonctions qui peuvent être réalisées à l'aide du système. Cette description va du général au particulier jusqu'à décrire les concepts du domaine impliqués par la réalisation des fonctions. Un second type de description consiste à décrire le système par l'énumération de ses parties constitutives. Tout comme la précédente, cette description suit une progression dans le degré de détail des parties. A un étage donné de la hiérarchie sont considérées des parties qui groupent des parties du niveau inférieur. Bisseret et Enard (1970) proposent de décrire un domaine en unités élémentaires de travail<sup>3</sup> selon deux logiques : la logique de construction des unités et la logique d'utilisation de ces unités. Richard (1983) différencie la logique de fonctionnement et la logique d'utilisation<sup>4</sup>. La première est une description causale fondée sur l'unité action-effet. La seconde description est relative aux actions à réaliser pour atteindre un but (logique de l'utilisation). Ces deux descriptions ne sont pas indépendantes<sup>5</sup> dans la mesure où les connaissances de fonctionnement permettent, si elles sont exprimées à un degré de détails suffisant, d'expliquer pourquoi une procédure est adaptée à un contexte particulier et, dans certains cas, de faire comprendre pourquoi une même action peut servir des

---

<sup>3</sup> Les auteurs différencient deux types d'unités élémentaires : les données et les processus.

<sup>4</sup> On pourrait également parler de logique de l'action.

<sup>5</sup> Ce qui ne signifie pas que l'une soit une condition nécessaire de l'autre. On peut en effet "avoir compris" et ne pas être en mesure "d'utiliser" et, inversement, "savoir utiliser" sans être en mesure de justifier l'utilisation en terme de fonctionnement (cf. Bisseret 1988).



objectifs différents. Par exemple, que l'action de "COUPER" du Macintosh puisse être utilisée pour déplacer une chaîne de caractères, mais également pour supprimer une chaîne puisque rien n'oblige l'utilisation d'une chaîne située dans le presse-papiers.

Baudet et Denhière (1989) envisagent la description d'un système sous trois points de vue :

- le point de vue transformationnel ;
- le point de vue téléologique ;
- le point de vue relationnel.

Le point de vue transformationnel permet de spécifier les transformations acceptables sur chacun des éléments du système. On y décrit les conditions de mise en œuvre des transformations ainsi que leurs effets sur les éléments sur lesquels elles s'appliquent. Les effets définissent les *événements* et la notion *d'action* renvoie à ce qui produit les *événements*. Une description de ce type peut être qualifiée de causale en ce sens qu'elle permet de remonter des effets aux causes. Les causes des effets sont les événements qui sont à l'origine de l'apparition des effets. Appliqué à la description d'une interface, ce point de vue revient à décrire le fonctionnement de l'interface sans mentionner les différentes utilisations que ce fonctionnement peut servir.

Le point de vue téléologique intègre le but visé lors de la réalisation d'une transformation. Ce type de description utilise le terme "pour" au sens de "si le but est de". Par exemple, "pour" atteindre un état "x", produire les effets "e1, e2, e3, e4.....". La description repose sur une décomposition du but en sous-buts de différents niveaux. La manière de réaliser les sous-buts est décrite par l'énumération des actions qu'il faut exécuter pour produire les effets souhaités. Il n'est pas difficile de voir que cette description est tournée vers l'utilisation et non vers l'explication des procédures. Ce dernier aspect est réalisé par le premier point de vue. Enfin le point de vue relationnel décrit les états du système à partir de l'énumération des objets qui les composent et des relations existantes entre ces objets.

En quoi ces travaux apportent des éléments constructifs pour le modèle ergonomique ? En premier lieu, si l'on considère l'utilisateur comme un exemple de la classe des systèmes de traitement d'informations, on peut le décrire selon les points de vue descriptifs valables pour la classe des systèmes. Ainsi, par la prise en

compte du point de vue téléologique, on peut espérer expliciter les connaissances<sup>6</sup> du sujet en termes de buts et de sous-buts. Ce résultat intermédiaire permettrait d'apporter des éléments d'appréciation du degré de compatibilité entre les buts et les sous-buts connus de l'utilisateur et les possibilités fonctionnelles de l'interface. C'est là une dimension de la compatibilité<sup>7</sup> à prendre en compte. Le travail de Kieras et Polson (1985) est une tentative dans cette direction. L'idée est de mesurer l'écart entre la structure des procédures connues pour atteindre un but et la structure des procédures impliquées par la réalisation du même but à l'aide d'une interface. Un graphe de transition d'états décrit l'interface. Il est transformé par un interpréteur en structure de buts. Les connaissances procédurales de l'utilisateur sont décrites par une structure de buts. La mesure retenue est le nombre de règles nécessaires pour passer de la description du graphe d'états à la structure de buts.

Cette approche est discutable pour plusieurs raisons. La première est que le mécanisme de transformation du graphe de description du dispositif à la structure de buts n'est pas décrit. Pour répondre à cette critique, une solution pourrait être de s'orienter vers l'utilisation de deux formalismes identiques pour les deux descriptions. La seconde critique tient au fait que le critère d'adaptation retenu (en terme de nombre de règles) pose un problème de validité dans la mesure où le nombre de prémisses dans la partie gauche des règles n'est pas pris en compte.

En second lieu, à partir du point de vue transformationnel, on peut espérer établir une comparaison entre les transformations subit par un objet lors de l'exécution d'une procédure sans l'interface et à l'aide de l'interface. C'est à notre avis une seconde dimension de la compatibilité qui peut être évaluée. A titre d'exemple considérons la situation suivante : je déplace un paragraphe d'un texte à l'aide d'une paire de ciseaux. Le premier sous-but consiste à rendre physiquement indépendant le fragment de la page sur lequel figure le paragraphe à déplacer. En vue de cela, je le coupe à l'aide des ciseaux. Le second sous-but vise à créer une dépendance entre le fragment isolé et le reste du texte. A cette fin, je mets en position le fragment à sa place définitive. Dans ces conditions, le fragment de texte déplacé a conservé la propriété d'être visible durant l'action. L'atteinte du même but à l'aide d'un traitement de texte peut entraîner l'absence de cette propriété : le fragment est caché dès lors qu'il est séparé du texte. Un examen systématique de ce type appliqué à l'ensemble des objets de l'interface permettrait d'apporter des

---

<sup>6</sup> Pour un domaine donné de tâches.

<sup>7</sup> On pourrait parler de compatibilité fonctionnelle.

éléments d'appréciation de la compatibilité. A ce titre, le point de transformationnel est à prendre en compte dans notre modèle.

Enfin le point de vue relationnel apporte des éléments pertinents pour évaluer les aspects structuraux des états (consistance de position, consistance de composition des objets...). De manière générale, ce point de vue doit révéler les cas où les attributs d'un objet (au sein d'un état) se différencient quand on passe d'un état à un autre. Si les différences portent sur des attributs qui participent à l'ergonomie de l'interface, leur identification facilitera l'évaluation.

La représentation unifiée de l'interface qu'on cherche à construire doit contenir les représentations venues des trois points de vue abordés ci-dessus. Il ne s'agit pas de dire que la prise en compte de ces points de vue assure une évaluation complète de l'interface. Il faudra les prolonger par des descriptions de plus bas niveaux. Cette réserve étant posée, il demeure tout de même que ces points de vue participent à l'évaluation de la compatibilité et de la consistance de l'interface.

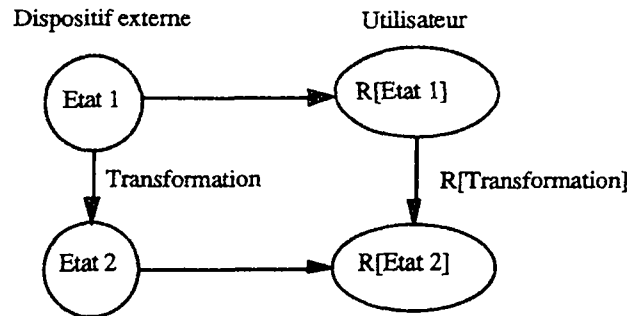
## 1.2 La description du système en termes de modèles mentaux

Une série de travaux s'est fixée comme tâche de décrire non plus les propriétés objectives de l'interface, mais plutôt l'intériorisation des propriétés par l'agent qui interagit avec le système. Selon cette orientation, Norman (1983) propose de différencier plusieurs variétés<sup>8</sup> de descriptions. La première est la description du modèle mental construit par l'utilisateur du système. La seconde est la description du modèle mental selon le point de vue du concepteur ; c'est le système dit conceptuel. La troisième variété a trait au modèle conçu par l'analyste pour rendre compte du modèle de l'utilisateur.

Pour chacun de ces modèles, on considère deux composants : les états et les transformations possibles sur ces états. La figure 1 présente le schéma général d'analyse.

---

<sup>8</sup> Le terme variété est utilisé, ici, à dessein pour bien marquer qu'il ne s'agit pas de classes définies à partir de l'énumération des propriétés définissant l'appartenance à la classe.

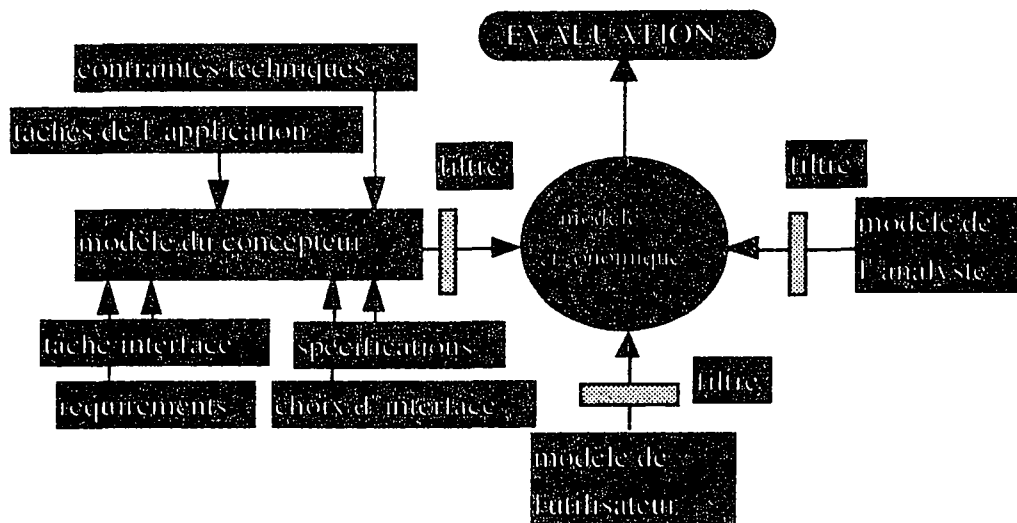


**Figure 1** : Schéma général d'analyse du modèle mental du dispositif chez l'utilisateur

L'analyse du modèle mental construit par le sujet est réalisée par l'explicitation de la représentation des états (notée R[Etat]) et de la représentation des transformations (notée R[Transformation]).

White et Frederiksen (1988, 1990) orientent la description des modèles mentaux vers une modélisation en termes qualitatifs venues des travaux de Kleer et Brown (1983). L'idée directrice s'énonce dans les termes suivants. La compréhension d'un système repose sur l'acquisition d'une succession de modèles causaux qui décrivent le système selon différentes perspectives. La première comprend ce que diSessa (1983) appelle les primitives phénoménologiques. Citons l'exemple des auteurs. La description causale d'un système électrique simple est décrite en terme d'action sur des commutateurs et d'effets visibles par l'agent de l'action,. (*j'actionne le commutateur ce qui a pour effet d'allumer la lampe qui lui est reliée*). A ce stade, l'agent rend compte des causes des effets visibles par l'action responsable de l'apparition de l'effet. Les perspectives suivantes mentionnent des entités abstraites pour rendre compte de l'effet visible (la différence de potentiel, la chaleur du filament de la lampe,...).

Les vues relatives aux modèles mentaux suggèrent le développement d'un modèle ergonomique de description. Ce modèle puise des informations venues des trois variétés de modèles proposées par Norman. La figure 2 exprime cette idée.



**Figure 2 :** Le modèle ergonomique et les modèles mentaux périphériques

On envisage le fonctionnement suivant. Une scrutation permanente des différentes descriptions au sein des modèles périphériques permet de sélectionner toute propriété pertinente<sup>9</sup> pour l'évaluation. Dès lors qu'une propriété pertinente est identifiée et qu'elle traverse le filtre, elle est capturée par le modèle ergonomique (cf. la toile d'araignée). Pour atteindre ce sous-but selon le fonctionnement proposé, il convient de faire vivre le modèle dans un contexte de conception clairement centré "évaluation ergonomique". Il est probable que les méthodes actuelles de conception d'application qui encouragent une évaluation terminale plutôt qu'intermédiaire rendent difficiles la création d'un tel contexte favorable au fonctionnement prescrit du modèle.

## 2 Les descriptions qui ont une fonction classificatoire sur les éléments de l'interface

### 2.1 Les descriptions fondées sur les niveaux d'interface

A côté des vues théoriques sur les variétés descriptives, des cadres descriptifs spécifiques aux interfaces informatiques se sont développés. Une notion fournit un apport important et permet de rendre compte de l'organisation d'une interface c'est celle de niveau d'interface. L'utilité de cette variété de description divise encore les chercheurs. A ce sujet, on consultera le texte de Scapin, Reynard et Pollier (1988) qui énumère les questions en suspens induites par la notion de niveau d'interface. Il n'en demeure pas moins qu'un large accord s'est réalisé sur l'organisation de ces

<sup>9</sup> A ce niveau, on considère qu'il est faisable de construire un inventaire des propriétés ergonomiques de l'interface (cf. Scapin 1990).

niveaux. Même si les auteurs se différencient par le nombre et la nature des niveaux introduits dans la description, ils admettent au moins implicitement l'idée de niveau et l'idée de hiérarchie entre les différentes strates descriptives.

L'idée sous-jacente à cette conception est d'exprimer la structure de l'interface à l'aide d'une suite de niveaux d'interface. On entend par niveaux un critère de répartition des éléments. A partir de ce postulat, deux directions de recherche se sont développées. La première voie est centrée exclusivement sur la description des niveaux à partir des éléments intrinsèques de l'interface. Il s'agit de décrire les propriétés lexicales, syntaxiques et sémantiques des constituants de l'interface. La seconde voie est caractérisée par l'introduction dans la hiérarchie d'éléments externes à l'interface. On voit apparaître dans les descriptions des niveaux relatifs aux connaissances antérieures de l'utilisateur essentiellement relatives aux tâches. L'idée est de mettre en rapport les objets et les actions impliqués dans l'interface avec les objets et les actions du domaine tels qu'ils sont décrits par les futurs utilisateurs de l'interface. Avant de passer à la présentation du détail de ces travaux, signalons que la comparaison détaillée des différents niveaux proposés par les auteurs est délicate dans la mesure où nul d'entre eux ne fournit un ensemble précis de caractéristiques qui permettraient de décider de l'appartenance d'un élément à un niveau d'interface. Ce disant, on peut tout de même présenter les niveaux en fournissant des exemples de répartition. Nous le ferons lors de la présentation des différents travaux.

Le premier travail dans la première voie est celui de Foley, Wallace, et Chan, (1974) repris dans Foley et Van Dam (1984). Quatre niveaux sont différenciés. Le niveau lexical rend compte des caractéristiques de "surface" de l'interface. Les auteurs donnent comme exemples la dénomination des commandes<sup>10</sup> et les attributs graphiques des objets. La syntaxe décrit les règles de composition des éléments de l'interface qui permettent de produire une séquences d'entrées comprises et exécutables par l'interface. Deux classes de règles sont considérées. Une première classe pour les règles de combinaison des entrées et une deuxième classe pour les règles de combinaison des sorties. Le niveau sémantique décrit les effets des actions. Le niveau conceptuel permet d'énumérer les objets et les transformations possibles sur ces objets. Foley, Mc Cormick et Bleser (1984)<sup>11</sup> ont proposé une démarche analogue à celle de Foley et Van Dam, mais en introduisant une

---

<sup>10</sup> Est-ce vraiment un trait de surface pour l'utilisateur dans la mesure où ce que l'on souhaite c'est que le choix de la dénomination ne soit pas arbitraire, mais suggère les effets qu'elle va produire.

<sup>11</sup> Travail cité dans Frohlich et Luff (1989).

description plus détaillée de l'interface. Le concepteur est invité à décrire - à l'aide d'un formulaire spécifique - les classes d'informations ayant trait : aux types d'objets de l'interface ; aux relations entre les types d'objets ; aux propriétés associées à chaque type d'objets et enfin aux opérations réalisables sur ces objets. L'idée des auteurs est de fournir au concepteur un support prévu pour décrire les objets de l'interface. Pratiquement, chaque objet entrant dans la composition de l'interface est documenté par une série de rubriques :

- le nom ;
- la définition ;
- les propriétés ;
- les actions exécutables sur l'objet.

La spécification centrée sur les objets est accompagnée d'une spécification de la sémantique des actions elle-même décrite à l'aide des rubriques suivantes :

- 1) la fonction : ce à quoi sert la fonction ;
- 2) les paramètres : les arguments de la fonction ;
- 3) les résultats : les effets produits ;
- 4) les erreurs : les erreurs possibles ;
- 5) le feed-back : l'information sur la connaissance des résultats ;
- 6) les effets marginaux ;
- 7) l'état final résultant tel qu'il est visualisé à l'utilisateur.

La méthode préconisée par Foley et al. (1984) a fait l'objet d'une évaluation empirique par Frolich et Luff (1989). La description des entrées sous la forme d'une grammaire B.N.F. pose des difficultés importantes. Pour répondre à ces difficultés, Frolich et Luff proposent d'utiliser le formalisme des réseaux de transitions pour spécifier les entrées. De plus, les auteurs introduisent un formulaire de description des éléments lexicaux d'entrées absent de la méthode préconisée par Foley et al. (1984).

La seconde voie est plus directement tournée vers la conception de systèmes adaptés aux connaissances antérieures des utilisateurs. Le modèle typique dans cette voie est le "Command Language Grammar" de Moran (1981). La seule idée novatrice par rapport à celle introduite par Foley et al. (1974) est d'introduire explicitement dans la description un niveau tâche. La nouveauté réside dans l'introduction du niveau tâche plutôt que dans la composante tâche classique en ergonomie.

C.L.G. est une grammaire qui permet de décrire un système informatique à l'aide de cinq abstractions :

- 1° la tâche ;
- 2° la sémantique ;
- 3° la syntaxe ;
- 4° l'interaction ;
- 5° les actions physiques.

Chaque niveau représente le système dans sa totalité, mais raffine le niveau immédiatement supérieur. L'intérêt de C.L.G. est qu'une description peut être considérée à la fois comme la description d'un cycle de conception et comme la description des connaissances requises pour utiliser l'interface.

La première étape<sup>12</sup> de C.L.G. consiste à définir les tâches correspondant aux besoins de l'utilisateur. Cette description<sup>13</sup> est indépendante du système. Elle est traduite en entités et opérations au niveau sémantique. A l'intérieur de ce niveau, Moran introduit une distinction, pour séparer les entités et les opérations propres au domaine des tâches et les entités propres au système. Cette séparation permet d'évaluer l'écart entre les connaissances antérieures des utilisateurs et les connaissances requises pour réaliser une tâche à l'aide du dispositif. Cette dichotomie est reprise dans un travail ultérieur de Moran (1983) qui vise à établir un critère quantitatif d'évaluation de l'écart entre les deux variétés de connaissances.

L'interaction entre le système et l'utilisateur est prise en compte au niveau syntaxique et de l'interaction. A ce niveau sont explicitées les relations entre la syntaxe du dialogue et les entités conceptuelles décrites dans les niveaux supérieurs. La description syntaxique prend en compte les contextes dans lesquels se déroulent les différentes phases du dialogue. Cette description des contextes permet au concepteur de structurer le dialogue et surtout d'expliciter les variables d'état à mémoriser par l'utilisateur lors du processus d'apprentissage du système.

Maints débats ont porté sur l'intérêt de C.L.G. La discussion a pris la forme d'une polémique stérile<sup>14</sup> sur tel ou tel aspect de C.L.G. Dans ce débat, la recherche de Sharrat (1988) est la seule qui apporte des faits expérimentaux qui permettent d'évaluer l'efficacité de ce cadre descriptif pour la conception. Le résultat

---

<sup>12</sup> Un exemple complet de description C.L.G est disponible dans l'article de référence de Moran (1981), mais également en français dans l'ouvrage de J.Coutaz (1990) aux pages 381 à 418.

<sup>13</sup> C'est la phase de "requirements analysis" introduite dans les méthodes de conception d'application informatique.

<sup>14</sup> Le lecteur pourra consulter le numéro 4 du volume 19 du SIGHI bulletin de l'année 1988.



important est que l'utilisation du cadre C.L.G. aboutit à une spécification comportant de maintes imprécisions essentiellement dans la définition des entités conceptuelles. A notre connaissance aucune recherche ne s'est fixée comme objectif d'étudier l'efficacité de C.L.G. pour l'évaluation.

Nielsen (1986) a proposé un modèle de l'interaction comprenant sept niveaux<sup>15</sup> de composition de l'interface. Ce cadre descriptif est plus directement axé sur l'adaptation de l'interface aux utilisateurs dans la mesure où il comprend un niveau spécifique réservé aux concepts issus de la description des connaissances antérieures du domaine telles qu'elles sont stockées dans la tête des gens du domaine. Ce niveau est appelé le niveau *but*. Il a pour fonction de recevoir les objets de connaissance du domaine tels qu'ils sont décrits par les opérateurs. Deux questions nous semblent devoir être posées. La première concerne la nature des connaissances introduites dans le niveau but. S'agit-il de connaissances expertes ou non expertes ? On peut envisager d'y répondre en introduisant au sein du niveau une organisation qui permettrait de différencier les deux variétés de connaissances. Une question plus difficile est de déterminer l'unité de connaissance introduite dans le niveau. L'unité but nous semble trop générale. Une solution pourrait être d'adopter une procédure de décomposition du but qui puisse permettre de définir un degré de raffinement de la description du but pour être en mesure de définir une unité de segmentation. Enfin un problème en suspens est de définir des règles d'appariement entre le niveau but et le niveau tâches. Cet aspect est crucial pour tendre vers la définition d'une distance entre les deux niveaux pouvant jouer le rôle de critère d'évaluation. Ces niveaux peuvent également avoir une fonction classificatoire sur les connaissances disponibles en ergonomie.

Par rapport à notre objectif, ces niveaux permettent d'envisager l'introduction d'une hiérarchie au sein de la représentation unifiée de l'interface. C'est là un atout important pour définir des niveaux d'évaluation.

## 2.2 Les descriptions des connaissances requises pour utiliser l'interface

### a) la description des connaissances sémantiques

Parallèlement aux recherches à caractère classificatoire, des travaux plus fortement marqués par l'approche cognitive tentent de proposer des méthodes de

---

<sup>15</sup> Nielsen a voulu être compatible avec la norme I.S.O. de conception de réseaux. Cette norme est fondée sur un modèle à sept niveaux.

descriptions des interfaces qui puissent permettre de prédire les difficultés cognitives liées à l'utilisation de telle ou telle interface.

Dans ce courant, une idée partagée par un grand nombre d'auteurs est de considérer qu'il faut éviter d'avoir à traiter deux descriptions différentes (logique causale et logique orientées buts et sous-buts) (Scapin et Pierret-Golbreich (1989) ; Hoppe, 1990 ; Poitrenaud, Richard, Denhière (1989)). C'est pour cette raison que l'on a commencé de voir apparaître dans la littérature des cadres descriptifs visant à intégrer les éléments relatifs aux tâches (buts et sous-buts connus) et au fonctionnement du dispositif. Ces deux variétés sont conceptualisées par (Halasz, (1984), Payne (1987), Young et Whittington (1990)) ; comme des espaces de résolution de problème. On parle ainsi d'espace de la tâche et d'espace du dispositif, Décrire la réalisation d'une tâche sur un dispositif revient alors à décrire les deux espaces et les règles de passage d'un espace vers l'autre espace.

Cette idée de recherche débouche sur une conception sémantique de l'adaptation des interfaces. La compatibilité s'exprime en termes de traits sémantiques communs au domaine des tâches et aux commandes de l'interface. La construction d'un indice de distance sémantique inter-domaines devient une exigence prioritaire (Moran, 1983). Ce disant, il faut bien garder à l'esprit que ces travaux en sont encore, à notre connaissance, à une phase exploratoire et qu'on est très loin d'avoir une solution définitive à cette question. Qu'en est-il des travaux qui tentent de répondre à cette question ? C'est ce que nous allons examiner dans la suite du texte.

A partir du travail de Reisner (1981a, 1981b, 1984), Payne et Green (1983), Green, Schiele et Payne (1988) ont proposé de décrire l'interaction comme une suite d'unités appelées Task Action Grammar. L'idée est d'aborder la description des tâches à partir de la notion d'unité sémantique primitive. Cela revient à injecter de la sémantique dans le travail de Reisner (1981a, 1981b, 1984). Une unité T.A.G. est assimilée à un tout composé de trois parties. Chacune des parties a une composante stable et une composante variable. Le premier plan de description porte sur la définition des traits sémantiques impliqués dans l'exécution d'une tâche. Ce plan est le plus abstrait. Il revient à décrire un univers sémantique. Le second plan de description spécifie les règles d'attribution de valeur aux attributs sémantiques. Il assure la construction de ce que les auteurs appellent le dictionnaire des tâches<sup>16</sup>.

---

<sup>16</sup> Le dictionnaire des tâches n'est rien d'autre qu'une combinaison de tâches exprimées sous la forme de couples attributs-valeurs acceptables.

Un troisième plan de description définit les transformations nécessaires pour passer de la tâche décrite dans le plan sémantique aux actions exécutables sur le dispositif. Ces transformations sont décrites comme des schémas de règles. Un exemple de descriptions de suite de commandes sous la forme de T.A.G. est présenté ci-dessous. La description est mixte dans la mesure où sont jointes les contraintes d'ordonnement procédurales.

### Exemple de T.A.G.

#### **Liste des commandes**

Déplacer le curseur d'un caractère vers l'avant : CTRL-C  
 Déplacer le curseur d'un caractère vers l'arrière : ESC-C  
 Déplacer le curseur d'un mot vers l'avant : CTRL-W  
 Déplacer le curseur d'un mot vers l'arrière : ESC-W

#### **Définition des T.A.G : dictionnaire des tâches simples**

Déplacer le curseur d'un caractère vers l'avant  
 Traits sémantiques : [direction = avant], [objet = caractère]

#### **Définition des règles de spécification d'actions**

Tâche composée des traits [direction, objet]  
 Structure de la commande = un symbole pour [direction] +<sup>17</sup> une lettre pour [objet]

Symbole pour le trait [direction] avec la valeur [avant] touche : "CTRL"  
 Symbole pour le trait [direction] avec la valeur [arrière] touche : "ESCP"  
 Lettre pour le trait [objet] avec la valeur [mot] : touche "W"  
 Lettre pour le trait [objet] avec la valeur [caractère] : touche "C"

Il y a dans cette tentative des éléments pour évaluer la consistance de l'interface. Cet aspect ne doit pas masquer le fait que ce formalisme a plus été conçu afin de tester des hypothèses cognitives sur l'apprentissage de la sémantique de commande simple que pour fournir un outil systématique de description de l'interface. Les développements récents des travaux venus de cette conception en témoignent. Les plus significatifs sont ceux de Young, R., Green, T.R.G., & Simon, T. (1989) Young et Whittington (1990). Ils prolongent l'apport de Payne et Green par la conception d'un système<sup>18</sup> qui permet de prédire les difficultés d'apprentissage de la sémantique des commandes pour un utilisateur donné. Le principe du système consiste à décrire la tâche à réaliser et les caractéristiques de

<sup>17</sup> "+" veut dire "et puis".

<sup>18</sup> Le système est fondé sur l'architecture du système S.O.A.R développé initialement par Laird (1984). S.O.A.R est basé sur le principe d'espaces de résolution de problème (espace des états et espace des opérateurs) qui sont le siège de la production des sous-buts. Les sous-buts sont produits; soit par une méthode faible; c'est-à-dire une méthode indépendante des connaissances du domaine, soit par une méthode spécifique au domaine.

l'interface pour les commandes impliquées par la réalisation de la tâche. Ces deux descriptions assurent la construction de l'espace de la tâche et l'espace du dispositif. La tâche du système est alors de produire les procédures convergentes vers l'état final spécifié et respectant les contraintes introduites lors de la définition de l'espace du dispositif.

Pour conclure, retenons que l'approche T.A.G pourrait être utilisée pour décrire le niveau sémantique introduit dans les travaux de Foley, Moran, Nielsen. Par ailleurs, comme le souligne Kellog (1987), cette approche permet d'apprécier certaines des propriétés de consistance de l'interface. En outre, on retiendra comme critères d'évaluation :

- le nombre de règles utilisées pour passer de la tâche générale aux actions physiques ;
- le nombre de traits sémantiques introduits dans la description du dictionnaire des tâches ;
- le nombre de règles communes à plusieurs tâches ;
- l'ordre de déclenchement des règles pour des tâches différentes impliquant les mêmes règles.

#### b) La description des connaissances procédurales requises pour atteindre un but

Une part importante de travaux se sont consacrés aux développements de descriptions qui puissent permettre de bâtir des prédictions sur les temps d'exécution des tâches réalisées à l'aide de l'interface. L'idée sous-jacente à ces travaux est d'aborder les tâches d'interaction à l'aide de catégories descriptives proches de celles utilisées dans les tâches manuelles (méthode des temps alloués). On décompose la tâche en sous-tâches elles-mêmes décomposées en actions élémentaires. On accorde des temps d'exécution à chaque composant puis à partir d'un modèle additif on est en mesure de calculer le temps global de réalisation d'une tâche.

Le travail de référence dans cette direction est celui de Card, Newell et Moran (1983). La première idée des auteurs est de proposer de décrire la réalisation d'une tâche sur un dispositif à l'aide de quatre descripteurs :

- buts ;
- opérateurs ;
- méthodes ;
- règles de sélection des méthodes.

Le but est un descripteur composite. Il se décompose à partir de l'énumération des propriétés visées par l'agent. Les opérateurs tels qu'ils sont définis par les auteurs sont les transformations les plus élémentaires des objets physiques et de la représentation de la tâche chez l'agent. Ils assurent par leur combinaison la construction de méthodes. Enfin l'opérateur de sélection permet de retenir une méthode d'atteinte du but lorsque plusieurs sont candidates.

La deuxième idée des auteurs est de considérer que les descriptions réalisées à l'aide des opérateurs s'organisent selon un axe abstraction-particularisation. Les auteurs proposent de différencier quatre paliers sur cet axe :

- la tâche ;
- la commande ;
- l'argument ;
- l'action physique.

Ce découpage permet de bâtir des prédictions sur les temps de réalisation des tâches aux différents stades de conception d'une interface. Ainsi, il est possible de faire des prédictions sans avoir préalablement défini les actions physiques élémentaires. Les auteurs proposent pour le niveau commande de considérer deux indices de calcul du temps de réalisation :

- l'estimation de la fréquence d'utilisation de la commande ;
- l'estimation du temps alloué à chaque commande.

Si l'ensemble des détails de l'interface est connu, il est possible de faire des prédictions plus précises par l'utilisation de la formalisation appelée Keystroke. Cette formalisation se fonde sur les équations<sup>19</sup> de calcul des temps élaborées par Embley, Lan, Leinbaugh, et Nagy (1978).

Malgré l'effort de modélisation, ce modèle ne permet pas de rendre compte des situations d'élaboration de procédure. Une description de ce type permet essentiellement de prédire le temps de réalisation d'une tâche. De plus, le fonctionnement cognitif sous-jacent est le suivant : chaque sous-but produit est traité sans interruption et dans un ordre fixe. Le premier sous-but présent en mémoire de travail est traité en premier. Selon ce fonctionnement, on ne tient pas compte des incidents susceptibles d'intervenir lors du traitement d'un sous-but. Par ailleurs on considère que l'ordre d'apparition des sous-buts dans la pile définit

---

<sup>19</sup> Le lecteur pourra se reporter au chapitre 8 de l'ouvrage de Card, Newell, Moran (1983).

l'ordre de réalisation. Or cette conception peut être en contradiction avec les caractéristiques de l'activité de l'utilisateur.

### c) La description du modèle conceptuel de l'interface

Dans les descriptions procédurales examinées ci-dessus, aucun élément ne porte sur le modèle conceptuel sous-jacent à l'utilisation. Par exemple, les informations visualisées à l'écran ne figurent pas dans ce type de description. En conséquence, ces formalismes ne permettent pas d'évaluer les propriétés ergonomiques de l'interface liées au fonctionnement (par ex. : les prompts, les feedbacks...).

Pour répondre à cet inconvénient, Tauber (1988) tente de définir un formalisme qui puisse permettre de décrire conjointement les procédures d'utilisation et le modèle de fonctionnement sous-jacent. Il reprend les travaux de Jackendoff (1983) sur la sémantique des changements de localisation et ceux de Payne et de Green (1983) et Green, Schiele et Payne (1988) sur les T.A.G.

Dans son approche de l'interface est décrite comme une machine virtuelle<sup>20</sup> qui doit permettre à l'utilisateur de comprendre la sémantique du domaine auquel appartient l'application. Cette séparation est astucieuse parce qu'elle permet de séparer explicitement la conception de l'interface de celle de l'application. Actuellement, les primitives descriptives permettent de rendre compte des changements de localisation des objets de l'interface. Les catégories descriptives utilisées sont : l'objet ; la place ; le chemin ; l'état ; l'événement et l'action. Ces catégories permettent de définir des unités appelées E.T.A.G. Par exemple, l'objet "presse-papiers" du Macintosh est décrit comme une [place] possible pour les [objets] qui appartiennent aux classes [chaînes] et [graphiques]. Le déplacement d'un objet [chaîne] vers l'objet [presse-papiers] est décrit comme un [événement] de la manière suivante : [Événement Aller ([objet = chaîne]) ([parcours] de [(Place [(objet = Presse-papiers))]. La description du dispositif en ces termes constitue le modèle conceptuel. Sur le plan de l'évaluation, cette approche permet d'établir une énumération des propriétés de consistance conceptuelle.

---

<sup>20</sup> Gaines et Shaw (1984) proposent un modèle de l'interaction fondé sur la notion de machine virtuelle. La machine virtuelle est constituée d'objets qui d'un point de vue de la logique de l'application ne sont pas nécessairement justifiés, mais qui en revanche facilitent l'apprentissage de la sémantique des commandes par l'utilisateur.

### 3 Les descriptions centrées sur le dialogue et fondées sur une décomposition fonctionnelle de l'interface

#### 3.1 La décomposition de l'interface en composants fonctionnels

Si les modèles généraux de description des interfaces ne fournissent pas un des outils précis pour la conception, il n'en va pas de même pour les formalismes proposés pour décrire dans le détail le dialogue. Plusieurs tentatives ont été proposées. Ces tentatives proviennent essentiellement d'informaticiens s'étant directement confrontés à la question de la description du dialogue pour la conception. Ce qui fait que dans ces premiers développements le critère de sélection d'un formalisme a été sa capacité à assurer la complétude de la conception de l'interface en négligeant l'intégration des propriétés ergonomiques. Le but visé était de produire une description exécutable par un dispositif afin d'automatiser la production. Grosso modo, on peut différencier trois variétés de formalismes. Les formalismes fondés sur la notion d'états et de transitions entre les états. Les formalismes dont la source est la grammaire B.N.F. et enfin des formalismes qui enrichissent le B.N.F en lui apportant une possibilité de spécification d'attributs de valeurs des éléments de la grammaire. Cette dernière variété comprend deux espèces de formalismes se différenciant par la place accordée aux connaissances antérieures de l'utilisateur. Le troisième formalisme est fondé sur la notion d'événement, Green (1985a, 1985b).

Pour mettre en œuvre ces formalismes, il est rapidement apparu utile de procéder à une décomposition fonctionnelle de l'interface. Ainsi on a vu apparaître dans la littérature des propositions de décomposition mieux cerner les différents composants d'un système interactif. Signalons que ces propositions ont été aussi établies pour s'adapter aux contraintes techniques de réalisation des interfaces. On retiendra la décomposition introduite par Green (1985a, 1985b). Ce choix traduit le consensus établi autour de cette décomposition<sup>21</sup>. Que propose Green ? Green propose de décomposer l'interface en trois parties fonctionnelles suivantes :

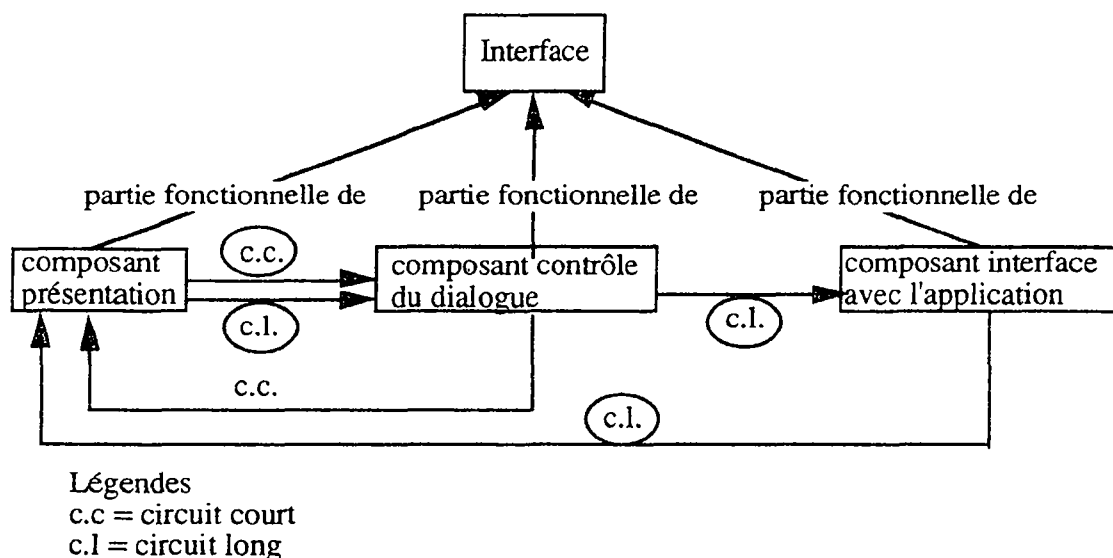
- 1) un composant présentation ;
- 2) un composant dialogue ;
- 3) un composant interface avec l'application.

---

<sup>21</sup> Cette décomposition a reçu le nom de modèle de Seeheim. Seeheim est le nom du lieu où s'est déroulé un workshop important sur les systèmes de gestion des interfaces utilisateurs. Le texte de El'Mrabet (1991) en donne une présentation détaillée.

Le composant présentation prend en charge la réalisation des transformations des objets graphiques de l'interface<sup>22</sup>.

Le composant contrôle du dialogue assure la mise en correspondance entre les entrées d'informations provenant de la présentation et les informations qui lui sont transmises par l'application. Pour décrire simplement le principe de fonctionnement<sup>23</sup> de ce module, il est commode de différencier deux modalités d'intervention. La première est réalisée par un circuit court : l'échange a lieu entre la présentation et le contrôle sans faire intervenir l'application. A l'inverse la seconde modalité est réalisée par un circuit long qui implique l'intervention des trois composants. En distinguant ces deux aspects, on comprend peut-être mieux pourquoi les auteurs informaticiens parlent d'indépendance du dialogue. C'est pour marquer la possibilité du module de contrôle à répondre à la présentation sans passer par un recours à l'application. C'est bien sûr une propriété qui a le statut d'objectif afin de décharger l'application de la gestion de la présentation. La figure 3 représente les composants de l'interface selon les vues décrites ci-dessus.



**Figure 3 :** Les composants de l'interface d'après le modèle de Seeheim

<sup>22</sup> Par exemple la transformation visuelle liée à la sélection d'un objet graphique au sein d'un écran de dialogue.

<sup>23</sup> Il faudrait à ce niveau de la présentation s'attaquer à la distinction entre dialogue synchrone et dialogue asynchrone ainsi qu'aux conséquences sur le fonctionnement du module de contrôle. Le texte de Hartson (1989) est tout à fait précis sur cette question.



### 3.2 La description du composant contrôle du dialogue

#### a) l'utilisation des grammaires B.N.F.

Pour décrire les séquences d'actions impliquées par la réalisation d'une tâche à l'aide de l'interface, une hypothèse de travail partagée par plusieurs auteurs a été d'adopter le formalisme B.N.F<sup>24</sup>. Ce formalisme permet de décrire l'ensemble des segments syntaxiquement corrects d'un langage à partir d'une série de règles de réécriture. On distingue dans une grammaire B.N.F. quatre composants :

- 1) un ensemble de segments de départ ;
- 2) un ensemble de segments terminaux ;
- 3) un ensemble de règles de réécriture ;
- 4) un ensemble d'opérateurs de séquencement.

Reisner est le premier auteur à avoir introduit ce formalisme pour décrire le dialogue impliquée par l'utilisation d'une interface. La visée de l'auteur est explicitement évaluative.

Dans cette perspective, Reisner énumère quatre indices d'évaluation dont la valeur peut être connue à partir d'une description B.N.F.

- le nombre total de signes terminaux ;
- pour chaque tâche, le nombre de segments terminaux ;
- le nombre de règles de réécriture nécessaire pour passer d'un segment de départ à un segment terminal ;
- l'ordre d'application des règles pour des tâches dont la décomposition met en œuvre les mêmes règles.

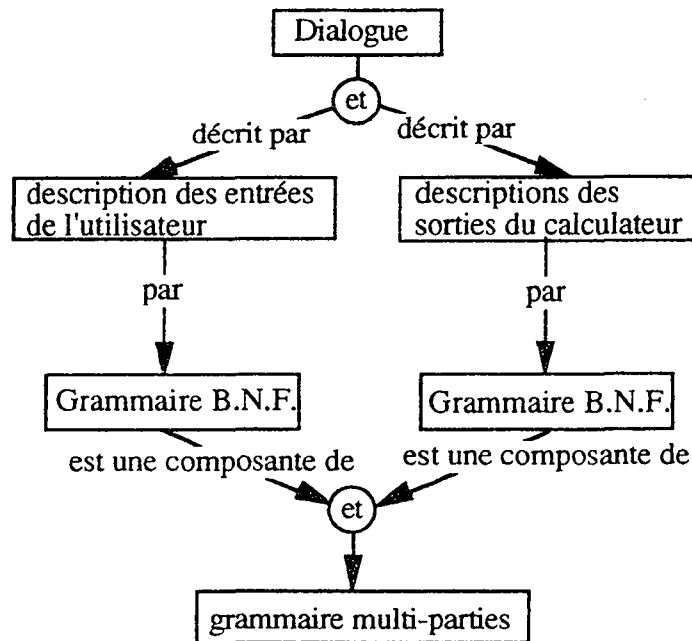
En accord avec Reisner (1981a, 1981b), Dunsmore (1982) propose des critères d'évaluation similaires.

Les descriptions en B.N.F. de Reisner n'intègrent pas les éléments relatifs au fonctionnement de l'interface. C'est une limite partagée avec les T.A.G. Pour répondre à cet inconvénient, l'idée de Shneiderman (1982) est d'introduire dans la description B.N.F. la spécification de la composante de l'interaction à l'origine du changement d'état du dialogue. La seconde idée est d'inclure dans la grammaire un descripteur lexical des affichages destinés à l'utilisateur (cf. figure 4). Ces descripteurs permettent de spécifier la configuration spatiale des écrans à partir d'une

---

<sup>24</sup> Acronyme de Backus-Naur-Form.

décomposition de ceux-ci en objet plus élémentaires. Les transformations réalisées sur les objets sont décrites par un jeu de règles B.N.F.



**Figure 4 :** Composition de la grammaire multi-parties

#### b) Les graphes de transition

Un autre formalisme utilisé pour décrire le dialogue est le graphe de transition Parnas (1969), Wasserman (1982), Jacobs (1983). Le dialogue est décrit comme une suite d'états reliés entre eux par des arcs correspondants à des inputs d'activation. Le graphe décrit les transitions de fonctionnement normal et les transitions liées à l'entrée d'inputs erronés en les reliant à un nœud Erreur. La procédure de récupération de l'erreur est décrite comme un retour à l'état initial. Un enrichissement consiste à introduire une hiérarchie de graphes afin d'éviter d'avoir à représenter sur une même représentation un nombre trop élevé de nœuds et de transitions entre nœuds. Cette technique a été systématisée par la création de la notion de graphe de transition généralisé, Kieras et Polson (1983), Kieras et Polson (1985), Wasserman (1985), Chi (1985).

Une autre modification apportée aux graphes consiste à introduire une capacité de rétention réalisée par des composants appelés registres. Chaque registre est indexé à partir de la désignation de la classe d'entrées qu'il peut recevoir. Par

exemple, un registre “entrées clavier” recevra et donc se souviendra uniquement des inputs définis par le concepteur comme des entrées clavier.

Wasserman (1985) a posé directement la question de l'intégration des propriétés ergonomiques dans la description de l'interface. C'est ainsi qu'il a énuméré cinq critères de mise au point d'une technique de description du dialogue :

- 1) disposer d'un formalisme ;
- 2) la complétude : la technique de description doit prendre en compte : les entrées utilisateurs, les sorties vers l'utilisateur, les relations entre les entrées et sorties et l'application ;
- 3) l'intelligibilité de la description pour assurer la compréhension de la description par une personne autre que son concepteur notamment et surtout par les utilisateurs futurs de l'interface ;
- 4) la flexibilité afin de permettre la description de plusieurs styles de dialogue ;
- 5) l'exécutabilité de la description.

Pour répondre à son objectif, Wasserman (1985) enrichit la description en considérant deux séries d'extension. La première consiste à introduire dans le graphe des éléments relatifs aux sorties du calculateur vers l'utilisateur. Par exemple, la position du curseur et la position d'un message affichée peuvent être spécifiées à partir de la déclaration de coordonnées spatiales : la spécification suivante [R12 C15 “choisir un item du menu”] permet de donner l'ordre d'afficher sur la 12<sup>e</sup> ligne et la 15<sup>e</sup> colonne de l'écran le segment de texte entre guillemets. Une seconde extension permet de spécifier les parties constitutives d'un écran de dialogue. Ces deux améliorations apportées permettent assurément de produire une description plus conforme aux exigences de l'activité d'évaluation.

L'application du graphe au composant présentation permet d'évaluer les parcours entre les différents écrans. Globalement, on peut dire que le graphe de transition permet d'avoir une idée précise des caractéristiques de l'espace théorique de problème (états et opérateurs) dans lequel va naviguer l'utilisateur. A partir de cet description, on peut évaluer des classes de propriétés relatives aux possibilités d'action exploratoire offertes à l'utilisateur (retour arrière, mémorisation des parcours....). On peut également évaluer la lourdeur (en terme de nombre d'états) des transactions.

#### 4. Les descriptions composites non formelles

Une série de travaux propose des éléments descriptifs qui sans être fondés sur des formalismes apportent des éléments à prendre en compte. Dans ce courant, Hix (1989) propose trois idées. La première idée est d'aborder la tâche de conception de dialogue comme la mise en relation d'objets, initialement conçus de manière indépendante. Cette conception repose sur une décomposition du dialogue en parties abstraites élémentaires appelées interactions. Une interaction est composée de cinq objets :

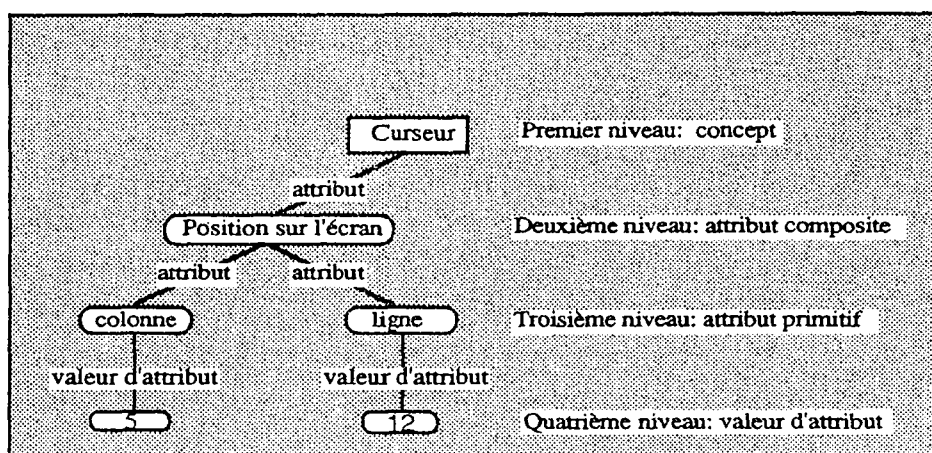
- un prompt : demande une entrée de la part de l'utilisateur ;
- un "end-user input" : mode de réalisation de l'entrée utilisateur ;
- une validation : module de contrôle de l'input ;
- une confirmation : information sur la justesse de l'input ;
- un "mapping" : traducteur de l'input en une valeur comprise par l'application.

La seconde idée est de représenter chaque interaction comme un tout composé de parties. Chaque partie est décrite à partir du listage de ces attributs. La troisième idée est de proposer au développeur des valeurs recommandées pour les différents attributs des parties. Le développeur peut alors accepter ou modifier chaque valeur. Après avoir déclaré les différentes valeurs, le développeur doit indiquer les relations qu'entretient l'interaction avec les autres composants. Une étude expérimentale a été réalisée pour tester l'efficacité de l'outil. L'expérience comprend une phase de création suivie d'une phase de modification d'interface. De cette manière, l'auteur établit que le temps de réalisation des tâches à l'aide de l'outil d'aide est réduit de plus de la moitié.

Le travail de Kitajima (1989) a pour objectif de bâtir un modèle qui permette de prédire la complexité cognitive de la réalisation d'une tâche à l'aide d'un dispositif. C'est une approche similaire à celle de Kieras et Polson (1985). L'auteur base son travail sur la mise en correspondance de la description de l'interface et de l'utilisateur. Nous présentons, ici, les éléments proposés pour décrire l'interface, les autres aspects ne rentrent pas directement dans notre propos.

Pour décrire le processus d'interaction Kitajima (1989) introduit la notion "*d'unité cognitive*". Une "*unité cognitive*" est un descripteur des objets qui ont un sens pour l'utilisateur dans le cadre de la réalisation d'une tâche. L'auteur adopte un mode de représentation des unités fondé sur quatre niveaux (cf. figure 5). Le premier niveau est le niveau conceptuel. Le second niveau permet d'explicitier les

attributs composites attachés aux concepts. La décomposition des attributs composites est réalisée au sein du troisième niveau. Enfin la valeur de chaque attribut est décrite par le quatrième niveau. La description du dispositif est réalisée à l'aide du formalisme des règles de production. La partie gauche contient les contraintes sur l'état du dispositif décrit comme une série de paires (ATTRIBUT-VALEUR). Six prédicats sont introduits pour décrire les contraintes sur les valeurs d'attributs [égal, non égal, plus grand que, moins que, après, avant]. Les actions sont décrites comme des changements des valeurs d'attributs.



**Figure 5 :** Description des objets de l'interface selon la procédure de Kitajima (1989)

Tyler et Treu (1989) se fixent pour objectif général de concevoir un module d'assistance à l'utilisateur qui assure un guidage selon deux volets :

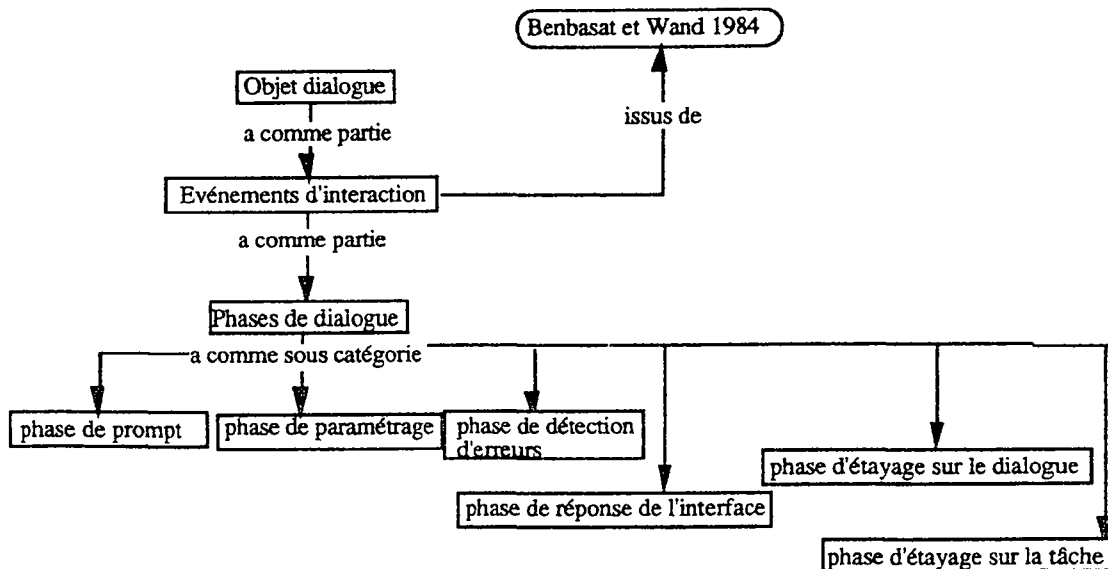
- un guidage relatif à la tâche ;
- un guidage ayant trait aux éléments de l'interface pertinents pour la réalisation de la tâche.

Pour atteindre cet objectif, les auteurs proposent une description originale du dialogue si on la compare aux vues antérieures. La première idée est de décomposer l'interaction entre l'utilisateur et l'interface selon une suite de phases présentant chacune des caractéristiques spécifiques.

Six phases sont différenciées : la phase de prompt ; de paramétrage ; de détection des erreurs ; de réponse de l'interface ; d'étayage sur le dialogue et d'étayage sur la tâche.

La conception de la partie présentation de l'interface est fondée sur cette décomposition du dialogue en parties. A chaque partie est associée une région de la

présentation. Le concepteur est ainsi amené à décrire tout événement d'interaction à l'aide des catégories de décomposition énumérées et d'y associer les régions de la présentation (cf. figure 6).



**Figure 6 :** Décomposition d'une unité d'interaction d'après Tyler et Treu (1989)

Cette approche est à retenir. En effet les critères de décomposition du dialogue en phases peuvent devenir des critères d'organisation des écrans de dialogue. On dispose ainsi d'un moyen systématique de découpage spatial de la surface de l'écran.

## 5 Conclusions et prolongements

Il ressort de ce texte plusieurs constats qui permettent de dégager des voies d'approfondissement. Pour dresser le bilan, on envisagera deux niveaux de conclusion. Un premier niveau général à partir des travaux sur les points de vue descriptifs. Un second niveau technique relatif aux formalismes à retenir.

### Premier niveau de conclusion

Par la prise en compte conjointe des points de vue descriptifs, on peut suggérer que le modèle unifié de l'interface que nous cherchons à construire comportera au moins trois classes d'informations. Premièrement, des informations relatives à l'architecture fonctionnelle de l'interface (but, sous-but, et peut-être commandes, actions). Deuxièmement, des informations portant sur les états de

l'interface liés à la réalisation des fonctions. A ce niveau, il faut séparer les états du composant présentation des états du composant dialogue. Ces états sont produits par des commandes ou par des actions considérées comme des parties des commandes. Troisièmement, des informations sur les transformations nécessaires pour passer d'un état à un autre.

## **Second niveau de conclusion**

Du point de vue où nous nous plaçons, le graphe de transition associé à une grammaire d'actions du type E.T.A.G. (éventuellement améliorée et "outillée" ) nous semble être une formule de base à retenir. Il permet de décrire la dynamique du dialogue à deux niveaux. En considérant le dialogue et la présentation comme deux systèmes à états séparés, on doit pouvoir générer un graphe logique et surtout un graphe décrivant les comportements des objets de la présentation. Ce sont là deux éléments importants pour l'activité d'évaluation.

Pour aller plus loin, il serait utile d'enrichir le formalisme par l'introduction de la dimension temporelle dans les graphes. On pourrait ainsi faire figurer les propriétés utiles à l'évaluation des temps de réponse et de l'invariance des temps. D'autres enrichissement plus élémentaires peuvent être envisagés. Par on peut également envisager d'associer à chaque état de la présentation un attribut pour spécifier le style de dialogue utilisé pour passer à l'état ultérieur.

Par ailleurs, avec le formalisme E.T.A.G. de description des transitions à retenir on disposerait ainsi de la sémantique et de la syntaxe des transitions attachées aux états et du modèle de fonctionnement sous-jacent (parties, interacteurs, prompt,...) de l'interface, impliquées dans l'exécution de la transition. Il reste à décrire la structure de chacun des états de dialogue.

L'idée serait de s'orienter vers la définition de primitives descriptives ergonomiques (P.D.E.). Une P.D.E. est dans cette perspective le plus petit morceau d'un état sur lequel on peut porter une évaluation. Sur le plan fonctionnel, ces P.D.E. sont susceptibles de déclencher au moins une règle d'évaluation.

L'ensemble de ces considérations est subordonné à la mise en œuvre d'une méthode de conception centrée "évaluation ergonomique" pour assurer tout au long de la conception une identification des propriétés nécessaires à la construction du modèle "Graph.Etag".

## Références bibliographiques

**Benbasat, I. & Wand, W. (1984)** A structured approach to designing human-computer dialogues, *International Journal of Man-Machines Studies*, (21) 105-126.

**Bisseret, A. (1988)** Modèles pour comprendre et réussir, In J. P. Caverni, C. Bastien, P. Mendelshon, G. Tiberghien ; *Psychologie cognitive, modèles et méthodes*, Presses Universitaires de Grenoble, 127-140.

**Bisseret, A. & Enard, C. (1970)** Le problème de la structuration d'un travail complexe, *Bulletin de Psychologie*, XXIII, 11/12, 632-648.

**Card, S., Newell, A. & Moran, T. (1983)** The psychology of human computer interaction, Lawrence Erlbaum Associates.

**Chi, U. H. (1985)** Formal specifications of user interfaces : a comparaison and evaluation of four axiomatic approaches, *I.E.E.E. Transactions on software engineering*, 11, (8) 671-685.

**Coutaz, J. (1990)** Interfaces homme-ordinateur : conception et réalisation, Dunod, 381-418

**diSessa, A. (1983)** Phenomenology and the evolution of intuition. In D. Gentner et A. Stevens (Eds.), *Mental Models*, Hillsdale, NJ : Lawrence Erlbaum Associates.

**Draper, S. W. (1986)** Display managers as the basis for user-machine communication. In D. A. Norman (Eds.), *User centered system design : new perspectives on human-computer interaction*. Hillsdale, New Jersey : Lawrence Erlbaum Associates Inc.

**Dunsmore, H. E. (1982)** Using formal approach as a design tool to predict the most useful characteristics of interactive systems, *Office Automation Conference Digest*, San Fransisco, A.F.I.P. Press, 53-56.

**deKleer, J. & Brown, J. S. (1983)** Assumptions and ambiguities in mechanistic mental models, In D. Gentner & A. Stevens (Eds.), *L.E.A.*



- El M'rabet, H. (1991)** Outils de génération d'interfaces : état de l'art et synthèse, Rapport technique N° 126. Rocquencourt I.N.R.I.A .
- Embley, D. W., Lan, N. T., Leinbaugh, D. W. & Nagy, G. A. (1978)** A procedure for predicting program editor performance from the users point of view, *International Journal of Man Machine Studies*, (10) 639-650.
- Foley, J. D., Mc Cormick, K. & Bleser, T. (1984)** Documenting the design of user-computer interfaces, (Tech. Rep. N° 14) Washington, DC : Computer Graphics Consultants, Inc.
- Foley, J. D. & Van Dam, A. (1984)** Fundamentals of interactive computer graphics, Addison-Wesley.
- Foley, J. D., Wallace, V. L. & Chan, P. (1974)** The human factors of computer graphics interaction techniques, *I.E.E.E. Computer Graphic and Applications*, 17, (6) 13-48.
- Frohlich, D. M. & Luff, P. (1989)** Some lessons from an exercise in specification, *Human Computer Interaction*, (4) 121-147.
- Gaines, B. R. & Shaw, M. (1984)** Principles of dialogue engineering. Nato Workshop on research needs in user-computer interaction. Loughborough.
- Green, M. (1985a)** Design notations and user interface management systems. In G. E. Pfaff (Ed.), : User interface management systems. Berlin-Heidelberg-New York ; Tokyo : Springer, 89-107.
- Green, M. (1985b)** Report on dialogue specification tools, In G. E. Pfaff (Ed.), : User interface management systems. Berlin-Heidelberg-New York ; Tokyo : Springer, 9-20.
- Green, T. R. G., Schiele, F. & Payne, S. J. (1988)** Formalisable models of user knowledge in human-computer interaction, In G. C. van de Veer, T. R. G., J. M. Hoc, D. Murray (Eds.), *Working with computers : Theory versus outcome*, Academic Press, 3-46.
- Halasz, F. (1984)** Mental models and problem solving in using a calculator, Ph.D, Department of psychology, Carnegie-Mellon University.

**Hartson, H. R. (1989)** User-interface management control and communication, I.E.E.E. Software, Janvier 1989, 71-77.

**Hartson, H. R. & Hix, D. (1989)** Toward empirically derived methodologies and tools for human-computer interface development, *International Journal of Man-Machines Studies*, (31) 477-494.

**Hix, D. (1989)** Developing and evaluating an interactive system for producing human-computer interfaces, *Behavior and Information Technology*, 8, (4) 285-299.

**Hoppe, H. U. (1990)** A grammar-based approach to unifying task-oriented and system-oriented interface descriptions, In D. Ackermann & M. J. Tauber (Eds.), *Mental models and human-computer interaction 1*, North-Holland, 353-374.

**Hurley, D. & Sibert, J. L. (1989)** Modeling user interfaces-application interactions, I.E.E.E. Software, Janvier 1989, 71-77.

**Jackendoff, R. (1983)** *Semantic and cognition*, Cambridge, M.A. : M.I.T. Press.

**Jacob, R. J. K. (1983)** Using formal specifications in a design of a human-computer interface, *Communications of the A.C.M.*, 259-264.

**Kellogg, W. A. (1987)** Conceptual consistency in the user interface : effects on user performance, In Bullinger, H. J. & Shackel, B. (Eds.), *Human-computer interaction*, Interact, 87, Amsterdam : North-Holland.

**Kieras, D. & Polson, P. (1983)** A generalized transitions network representation for interactive systems, In *Proceeding C.H.I. Human Factors in Computing Systems*, 103-106.

**Kieras, D. G. & Polson, P. (1985)** An approach to the formal analysis of user complexity, *International Journal of Man-Machines Studies*, (22) 365-394.

**Kitajima, M. (1989)** A formal representation system for the human-computer interaction process, *International Journal of Man Machines Studies*, (30) 669-696.

**Laird, J. E. (1984)** *Universal Subgoalng*, Department of Computer Science, Carnegie-Mellon University, CMU-CS-84-129, 1-60.

**Löwgren, J. (1988)** History, state and future of user interface management systems, S.I.G.H.I. Bulletin, 20, (1) 32-44.

**Moran, T. P. (1981)** The command language grammar : a representation for the user interface of interactive computer systems, International Journal of Man Machines Studies, (15) 3-51.

**Moran, T. (1983)** Getting into a system : external-internal task analysis. In A. Janda (Ed.), Proceedings of Human Factors in Computing Systems, A.C.M. Press, 45-49.

**Nielsen, J. (1986)** A virtual protocol dialogue for computer-human interaction, International Journal of Man-Machines Studies, (24) 301-312.

**Norman, D. A. (1983)** Some observations on mental models, In D. Gentner, A. L. Stevens, (Eds.), Mental models, London, Lawrence Erlbaum, 7-14.

**Parnas, D. L. (1969)** On the use of transition diagrams in the design of a user interface for an interactive computer system, Proceeding of the 24th ACM Conference, 369-385.

**Payne, S. J. (1987)** Complex problem spaces, modelling the knowledge needed to use interactive devices, Proceeding of interact' 87, North-Holland, 203-209.

**Payne, S. J. (1989)** A notation for reasoning about learning. In J. Long & A. Whitefield (Eds.), Cognitive Ergonomics and Human-computer interaction, Cambridge University Press, 134-165.

**Payne, S. J. & Green, T. R. G. (1983)** The user's perception of the interaction language : a two-level model. In proceedings CHI'83 Human Factors in Computer Systems, 202-206.

**Poitrenaud, S., Richard, J. F. & Denhière, G. (1989)** La description des procédures : une approche "orientée-objet", Rapport de Recherche, C.N.R.S - Université Paris 8, U.R.A. 218 : Psychologie cognitive du traitement de l'information symbolique.

**Pollier, A. (1990)** Etude descriptive d'une activité de conception-évaluation ergonomique d'un système de messagerie vocale.

**Rasmussen, J. (1985)** The role of hierarchical knowledge representation in decision making and system management, I.E.E.E. Transactions on systems, man, and cybernetics, 15, (2) 234-243.

**Reisner, P. H. (1981a)** Formal grammar and human factors design of an interactive graphic systems. I.E.E.E. Transactions on Software Engineering, 7, (2) 229-240.

**Reisner, P. H. (1981b)** Further developments towards using formal grammar as a design tool, C.H.I. 304-308.

**Reisner, P. H. (1984)** Formal grammars as a tool for analysing ease of use : some fundamentals concepts. In Thomas, J. C. & Schneider, M. L. (Eds.), Human Factors in Computer Systems. Norwood (N.J) : Ablex Publishing Corporation.

**Richard, J. F. (1983)** Logique de fonctionnement et logique de l'utilisation, Rapport de recherche n° 202 Rocquencourt, I.N.R.I.A.

**Scapin, D. L. (1990)** Organizing human factors knowledge for the evaluation and design of interfaces, International Journal of Human-Computer Interaction, 2 (3) 203-229.

**Scapin, D. L. & Pierret-Golbreich, C. (1989)** MAD : Une méthode analytique de description des tâches, Actes du colloque l'ingenierie des interfaces homme-machine, Sophia-Antipolis, I.N.R.I.A.

**Scapin, D. L., Reynard, P. & Pollier, A. (1988)** La conception ergonomique d'interfaces : problèmes de méthode, Rapport de recherche n° 957, Rocquencourt, I.N.R.I.A.

**Schiele, F. & Green, T. R. G. (1990)** H.C.I. formalisms and cognitive psychology : the case of task-action grammar. In M. Harrison et H. Thimbleby (Eds.), Formal methods in human-computer interaction, Cambridge University Press, Cambridge, 9-62.

**Sharrat, B. (1988)** Is C.L.G. usable ? S.I.G.H.I. Bulletin, 19, (4) 38-39.

**Shneiderman, B. (1982)** Multiparty grammars and related features for defining interactive systems, I.E.E.E. Transactions on Systems, Man, and Cybernetics. 12, (2) 148-154

**Tauber, M. (1988)** On mental models and the user interface, In van der Veer, G. C., Green, T. R. G., Hoc, J. M. & Murray, D. M. (Eds.), Working with computers : theory versus outcome, London : Academic Press.

**Tyler, S. W. & Treu, S. (1989)** An interface architecture to provide adaptative task-specific context for the user, International Journal of Man-Machines Studies, (30) 303-327.

**Wasserman, A. I. (1982)** The user software engineering methodology : an overview, In Information System Design Methodologies -A comparative Review, A. A. Verrijn-Stuart, (Ed.), Amsterdam, North-Holland, 591-628.

**Wasserman, A. I. (1985)** Extending state transition diagrams for the specification of human-computer interaction, I.E.E.E. transactions on software engineering, Vol. S. E., (8) 699-713.

**White, B. Y. & Frederiksen, J. R (1988)** Explorations in understanding how physical systems work, Proceeding of the tenth annual conference of the cognitive science society, Lawrence Erlbaum Associates, 325-332.

**White, B. Y. & Frederiksen, J. R. (1990)** Causal model progressions as a foundation for intelligent learning environments, Artificial Intelligence, (42) 99-157.

**Young, R. M. (1983)** Surrogate and mapping : two kinds of conceptual models for interactive devices. In mental models, Gentner et Stevens (Eds.), Hillsdale, N.J. : L.E.A., 35-52.

**Young, R. M., Green, T. R. G. & Simon, T. (1989)** Programmable user models for predictive evaluation of interfaces design, In K. Bice & C. Lewis (Eds.), Proceedings of Human Factors in Computing Systems, New York, 15-19.

**Young, R. M. & Whittington, J. (1990)** Using a knowledge analysis to predict conceptual errors in text-editor usage, Proceedings of Human Factors in Computing Systems, Seattle, 91-97.

## Index des auteurs

Baudet .....	3	Nagy .....	15
Bisseret .....	2	Newell .....	14
Bleser .....	8	Nielsen .....	11 ; 14
Card .....	14	Norman .....	5
Chan .....	8	Parnas .....	20
Chi .....	20	Payne .....	12
deKleer et Brown .....	6	Poitrenaud .....	12
Denhière .....	3 ; 12	Pollier .....	7
diSessa .....	6	Polson .....	4 ; 20 ; 22
Dunsmore .....	19	Rasmussen .....	2
Embley .....	15	Reisner .....	12 ; 19
Enard .....	2	Reynard .....	7
Foley .....	8 ; 9 ; 14	Richard .....	2 ; 12
Frolich .....	9	Scapin .....	7
Green .....	12 ; 13 ; 17	Scapin et Pierret-Golbreich ..	12
Halasz .....	12	Schiele .....	12
Hix .....	22	Sharrat .....	10
Hoppe .....	12	Shneiderman .....	19
Jackendoff .....	16	Simon .....	13
Kellog .....	14	Tauber .....	16
Kieras .....	20 ; 22	Treu .....	23
Kitajima .....	22	Tyler .....	23
Lan .....	15	Van Dam .....	8 ; 9
Leinbaugh .....	15	Wallace .....	8
Luff .....	9	Wasserman .....	20 ; 21
Mc Cormick .....	8	Whittington .....	12 ; 13
Moran .....	9 ; 10 ; 14	Young .....	12 ; 13

119

120

121

122

123

124

125

126

127

128



**ISSN 0249-6399**